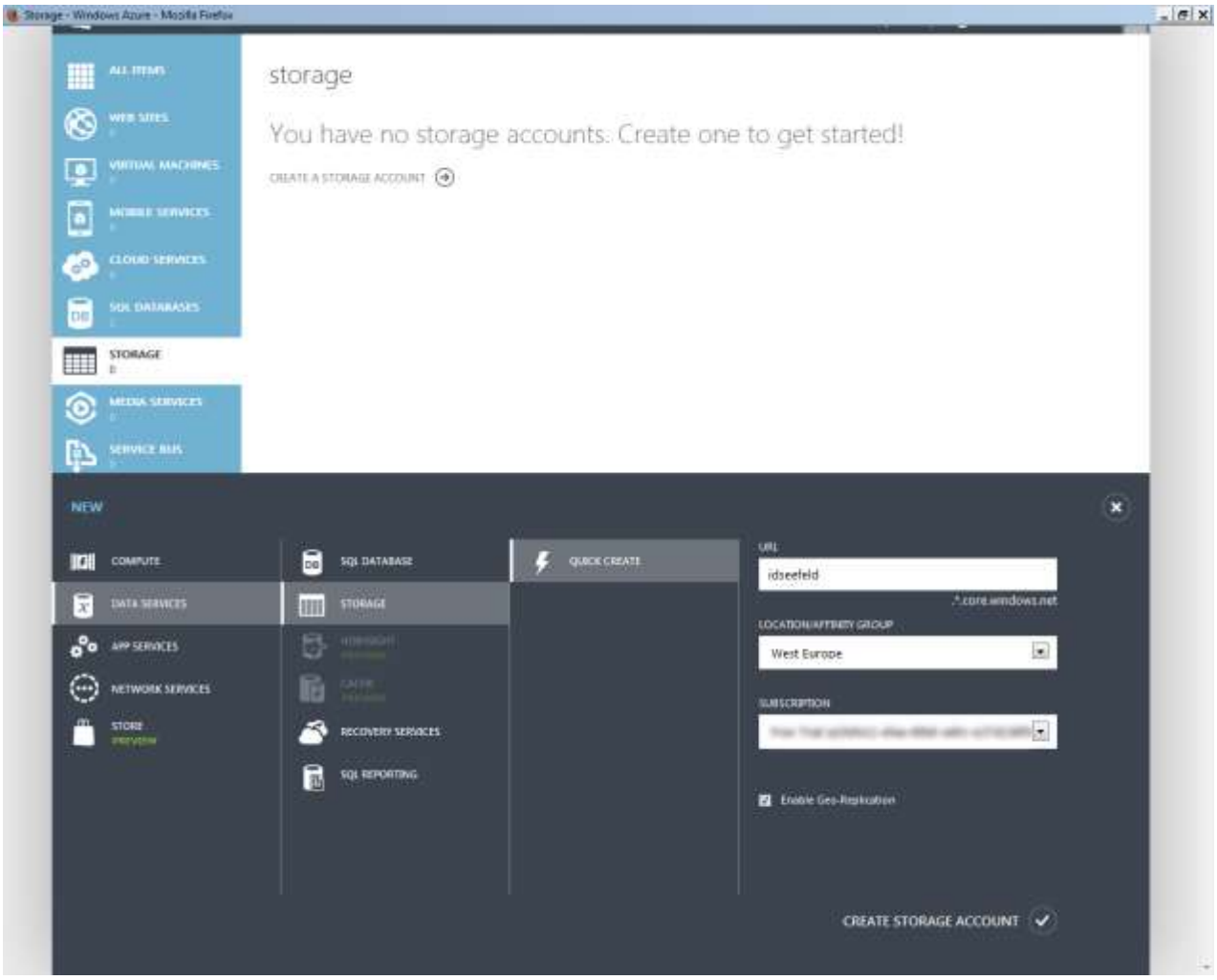
Azure Blob Storage Provider Documentation

What is the Azure Blob Storage Provider for Umbraco?

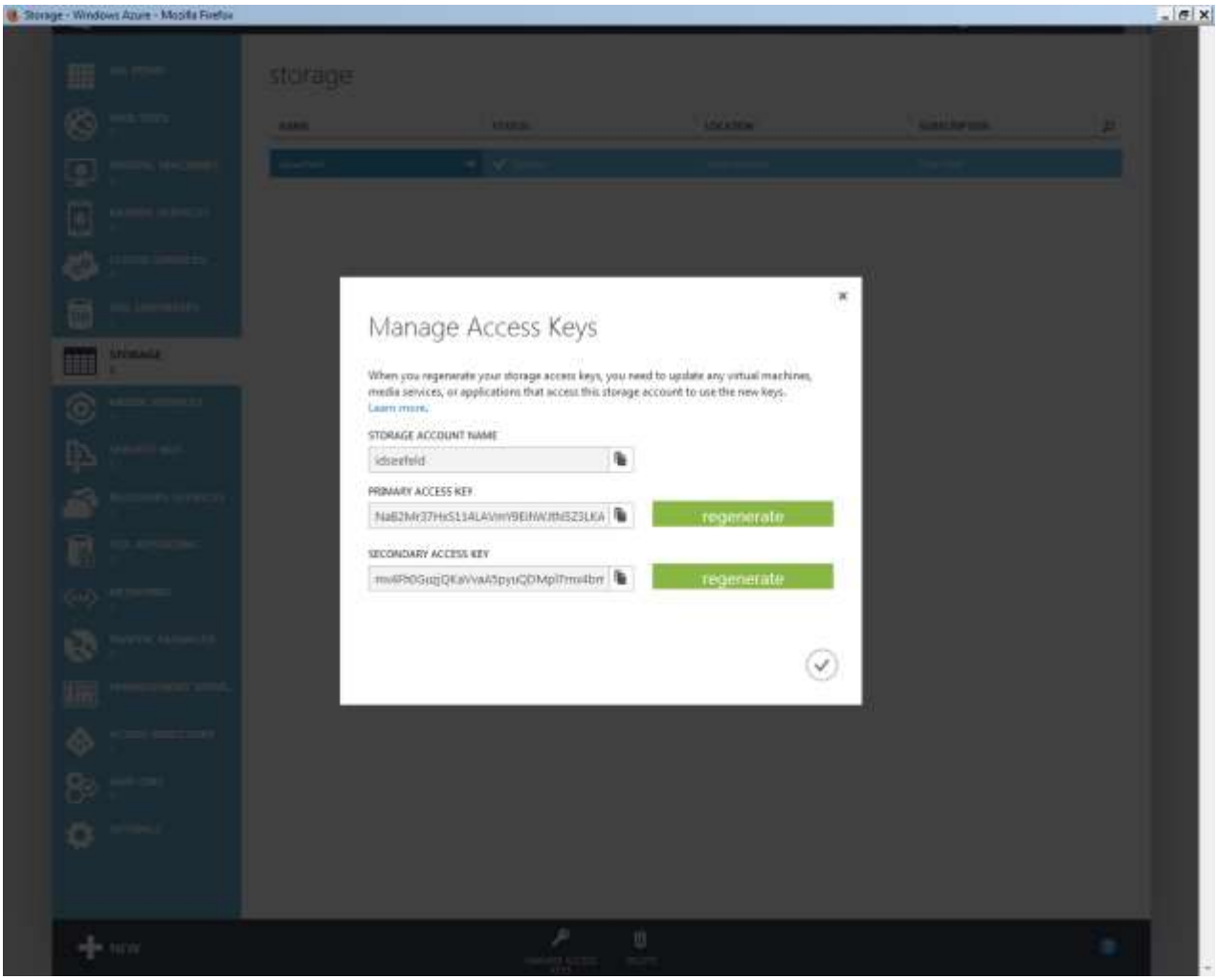
The Azure Blob Storage Provider replaces Umbracos default provider for media files. ***Full trust is required!***
The following steps show how to setup the *ABSP*:

1. At first you create the Azure storage.

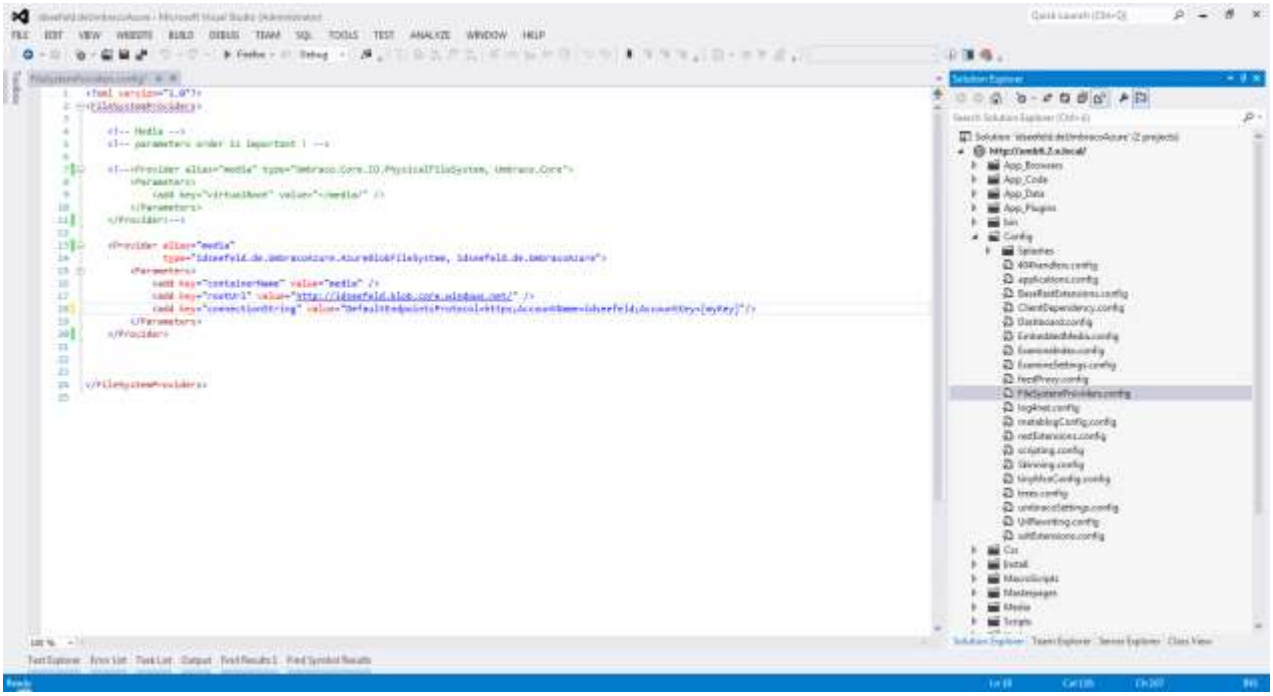


The Result

2. Copy one of the two generated keys.



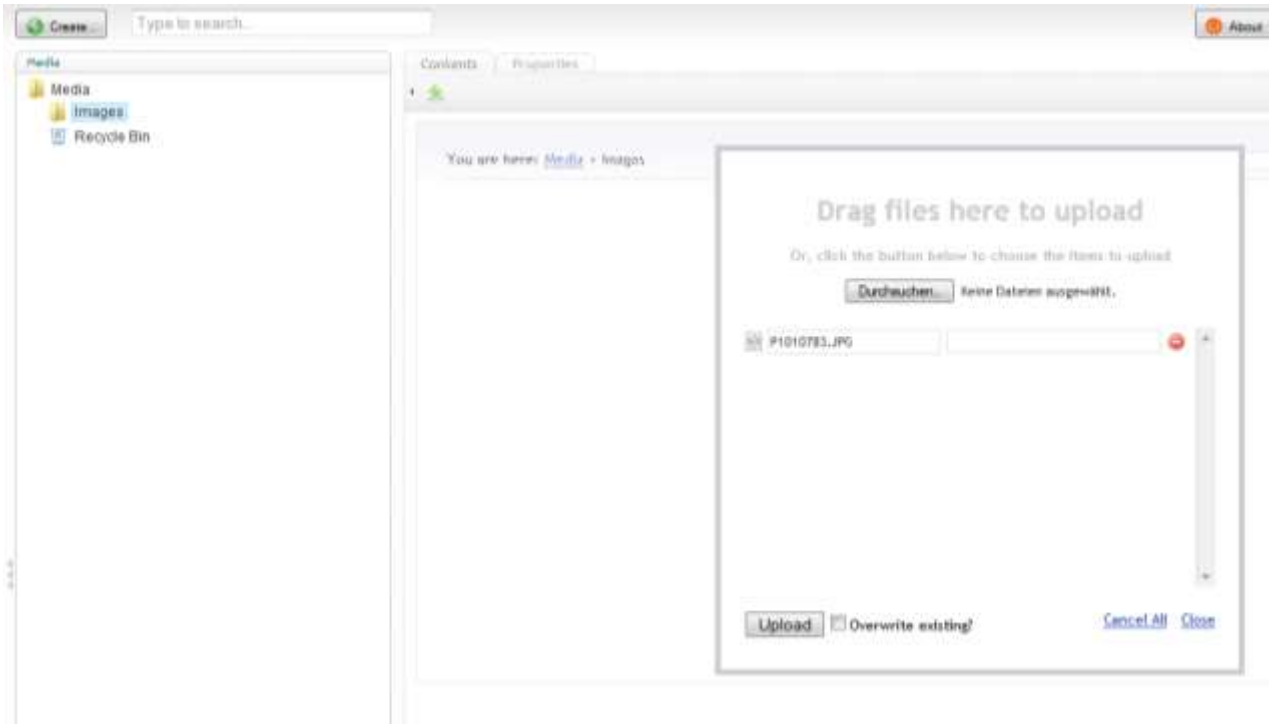
3. Open file ~/Config/FileSystemProviders.config of your Umbraco installation and paste your account name and key into the configuration:

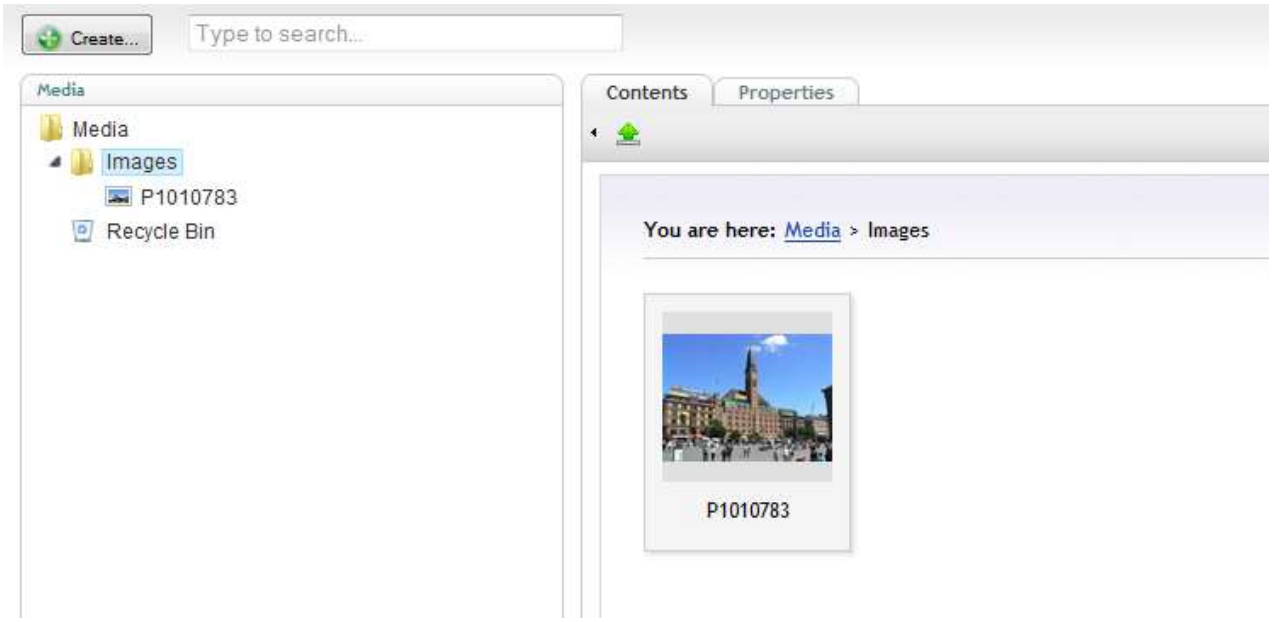


```
<?xml version="1.0"?>
<FileSystemProviders>
    <!-- Media -->
    <!-- parameters order is important ! -->

    <Provider alias="media"
                          type="idseefeld.de.UmbracoAzure.AzureBlobFileSystem,
idseefeld.de.UmbracoAzure">
                <Parameters>
                        <add key="containerName" value="media" />
                        <add key="rootUrl" value="http://[myAccountName].blob.core.windows.net/" />
                        <add key="connectionString"
value="DefaultEndpointsProtocol=https;AccountName=[myAccountName];AccountKey=[myAccountKey]"/>
                        <add key="mimetypes" value="woff|application/x-font-woff;mp3|audio/mpeg3" />
                </Parameters>
        </Provider>

</FileSystemProviders>
```

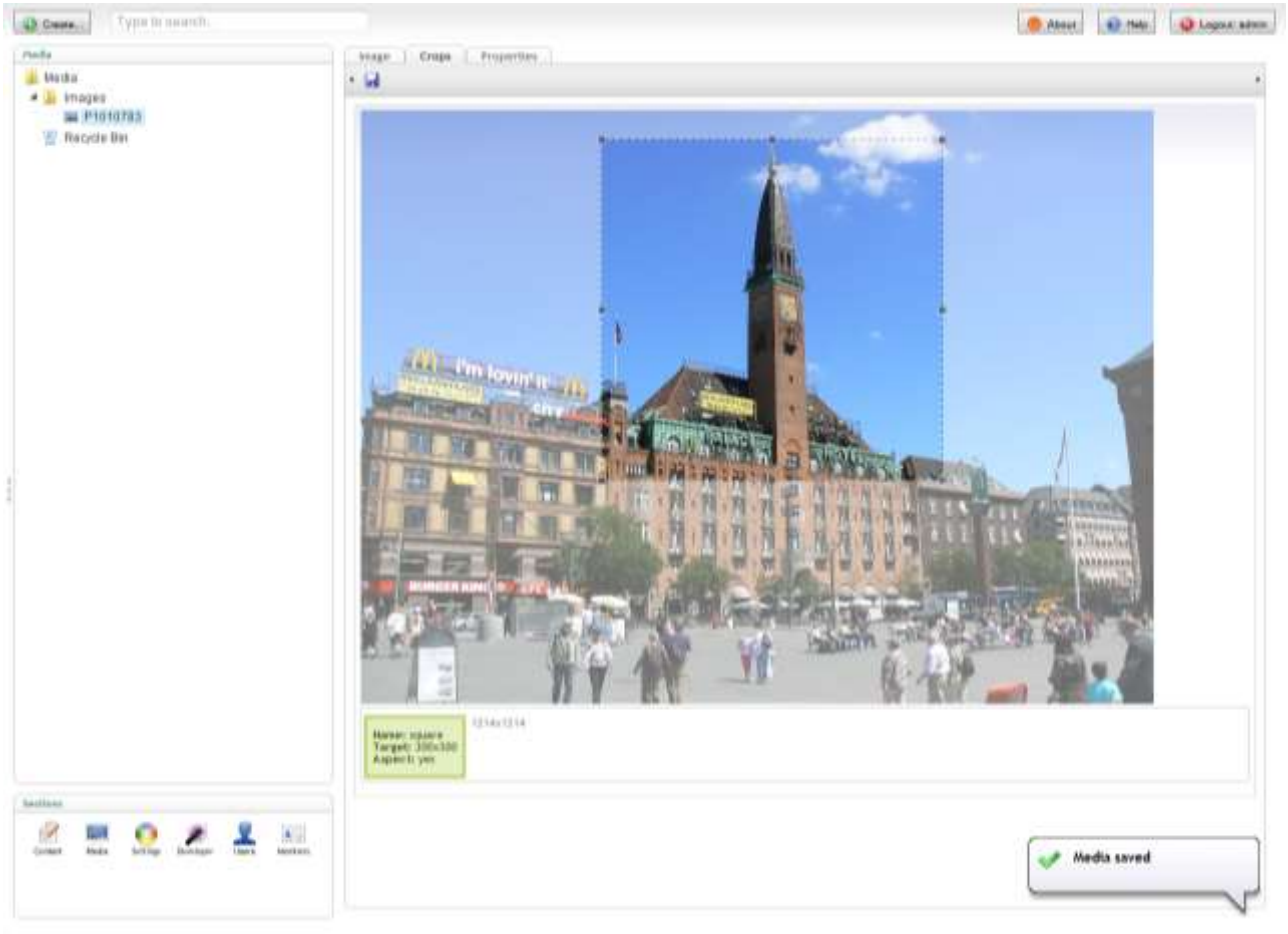
Since package version 1.0.7 you can add mimetypes. Separate each type description by ; (semicolon) and file extension (without leading dot) form header string by | (pipe). The following file extensions / types are *registered* by default: jpg, jpeg, gif, png, pdf and air. But you can override these if you like.

4. Restart the web server (e. g. touch web.config) and upload an image.

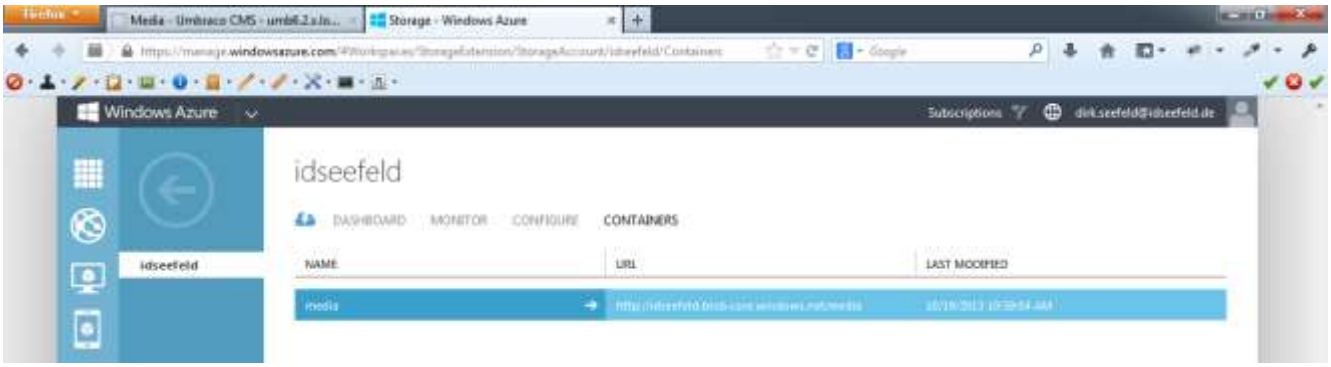




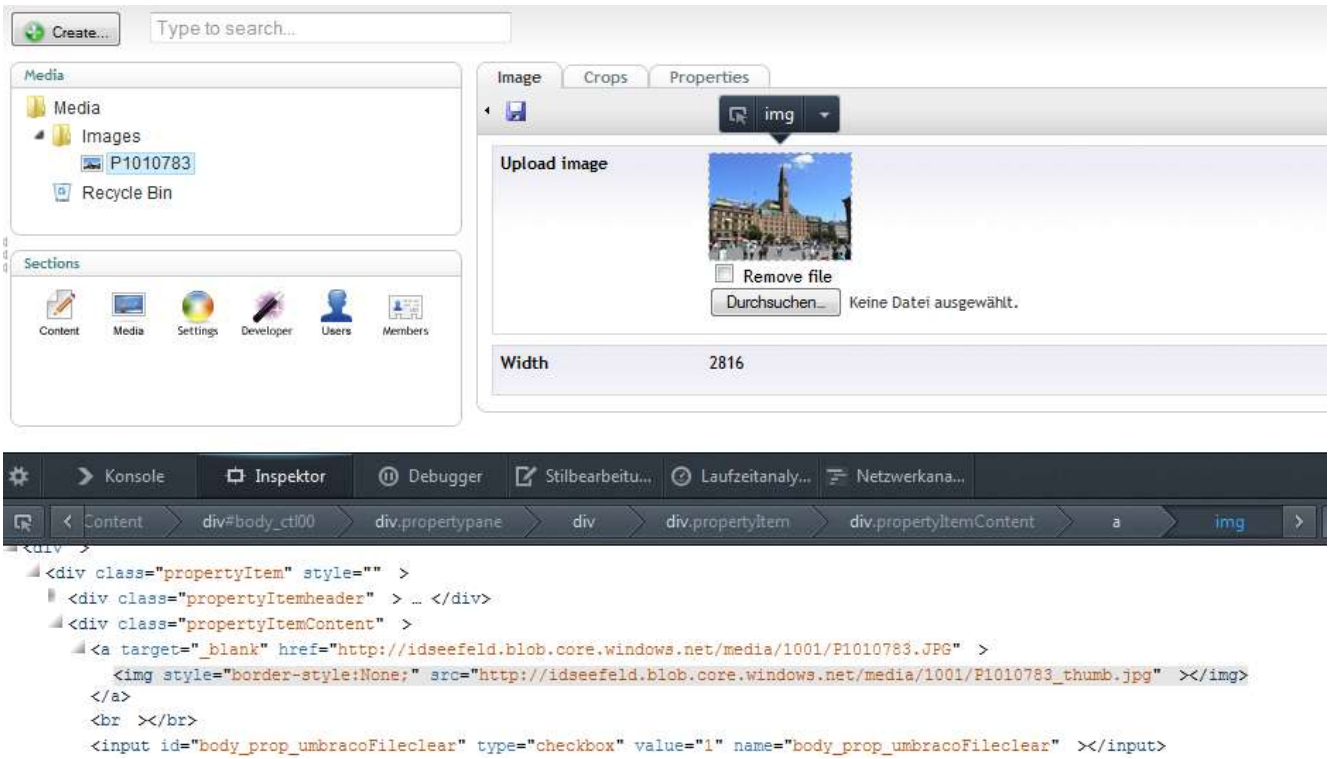
In my example I have previously installed the Image Cropper Extended and defined crops for media and content (for details see Image Cropper Extended documentation). *Just to show that it works!*



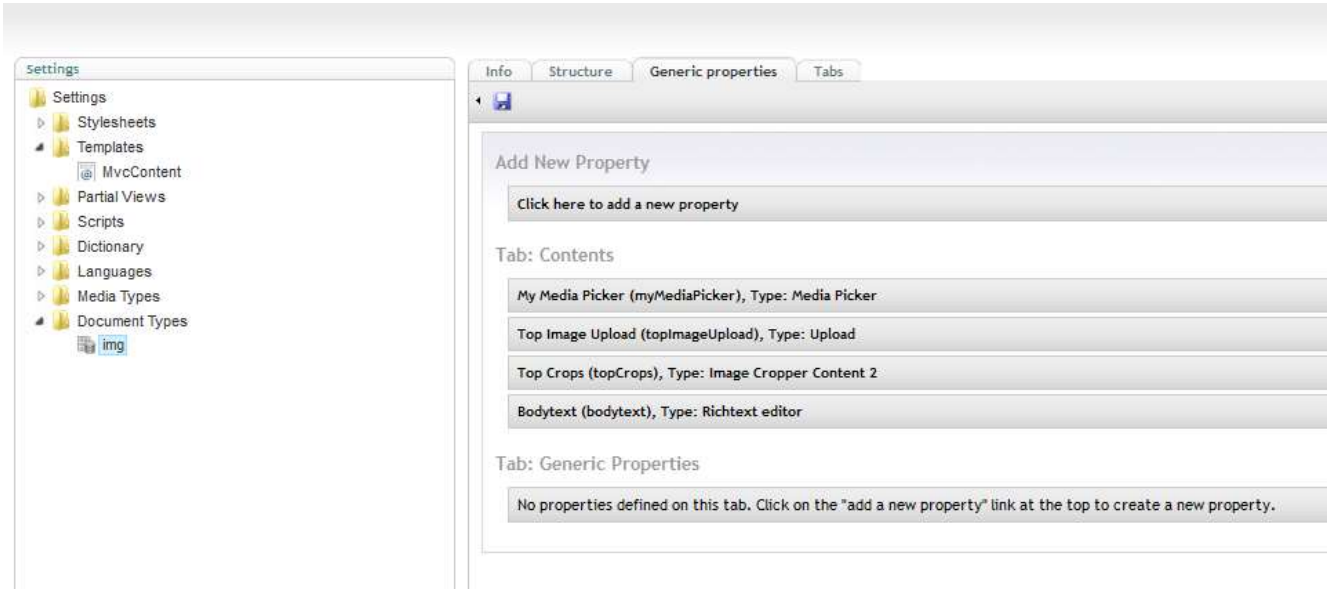
5. Check that your uploaded images are in the cloud ...



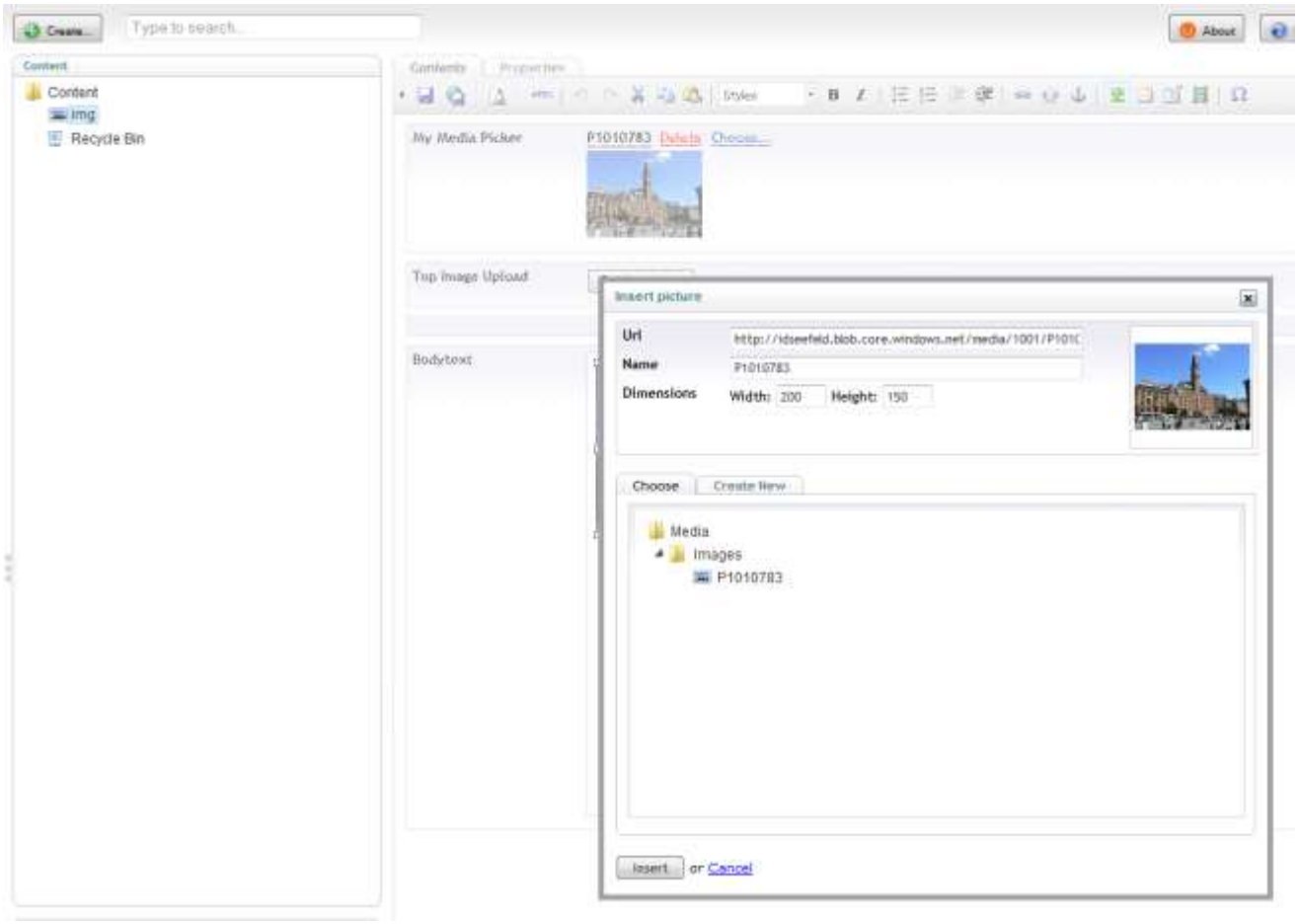
6. ... or explore the html.

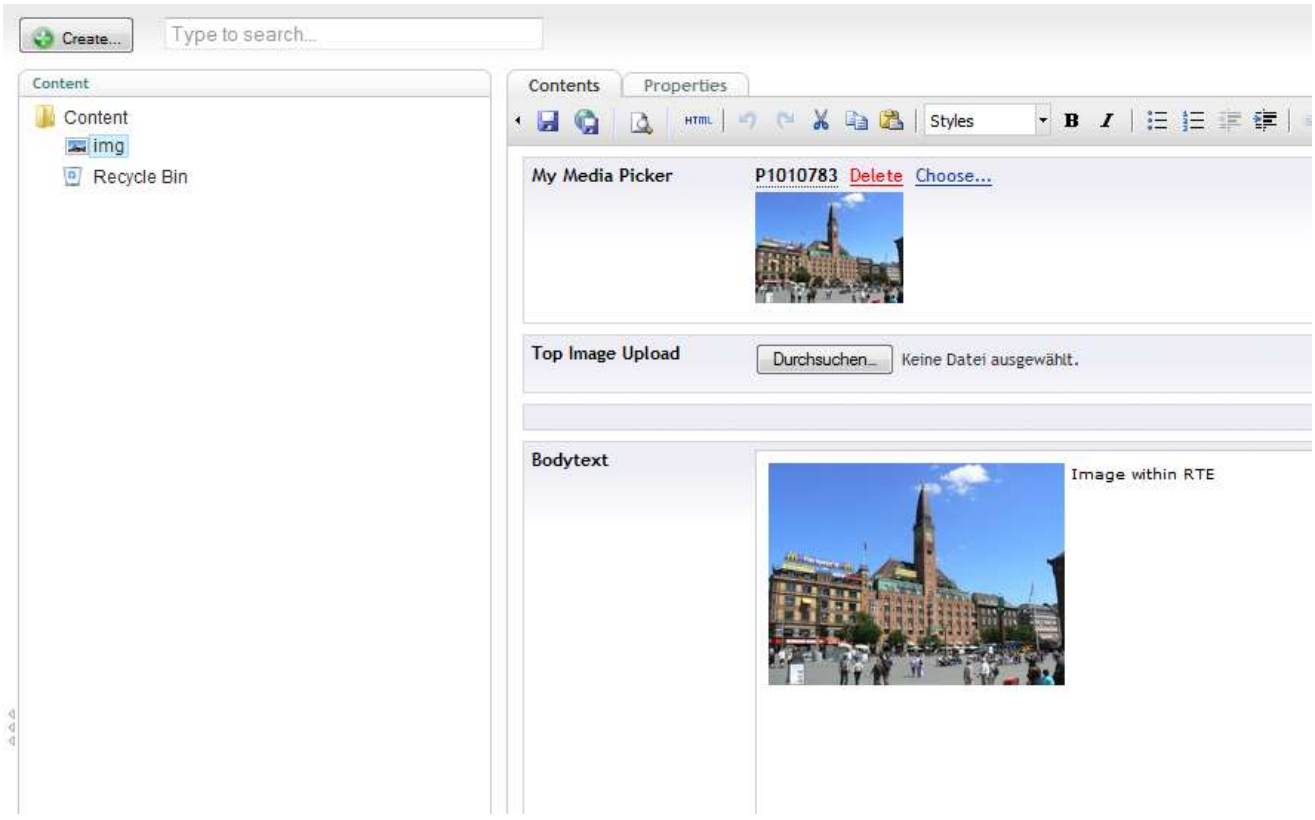


7. Example of a document type

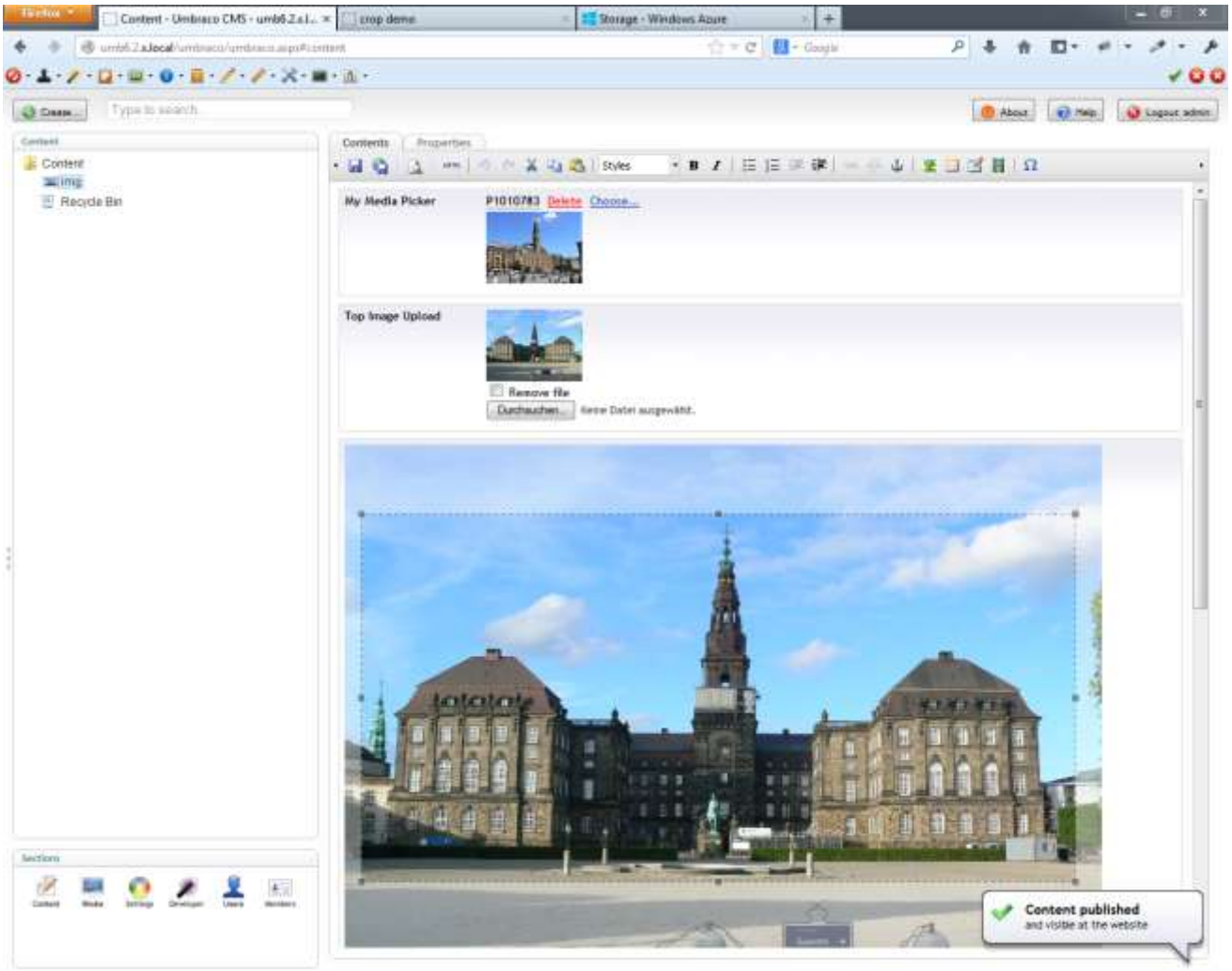


8. Create a node of the *img* document type and choose an image with the media Picker and insert one into the rich text editor.

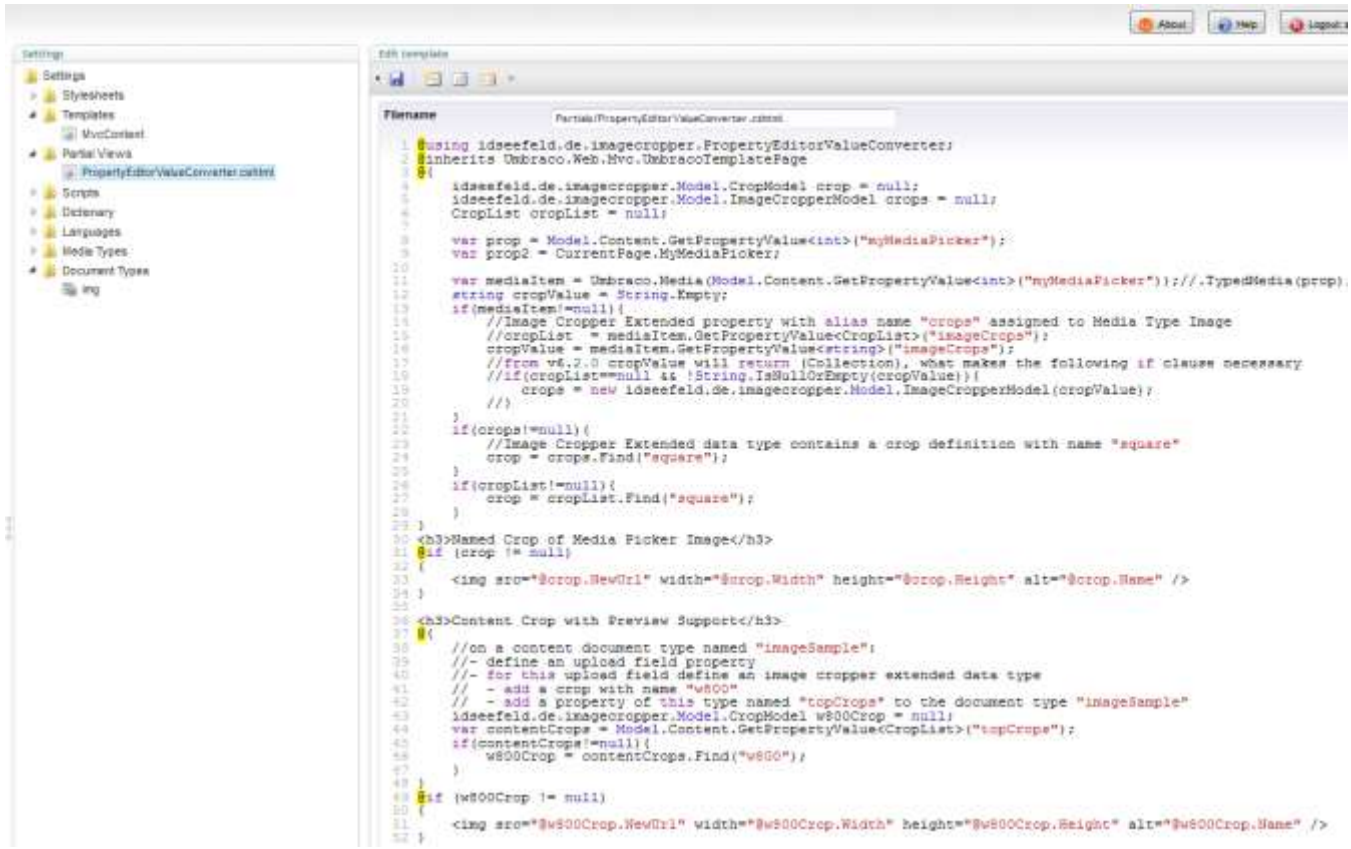




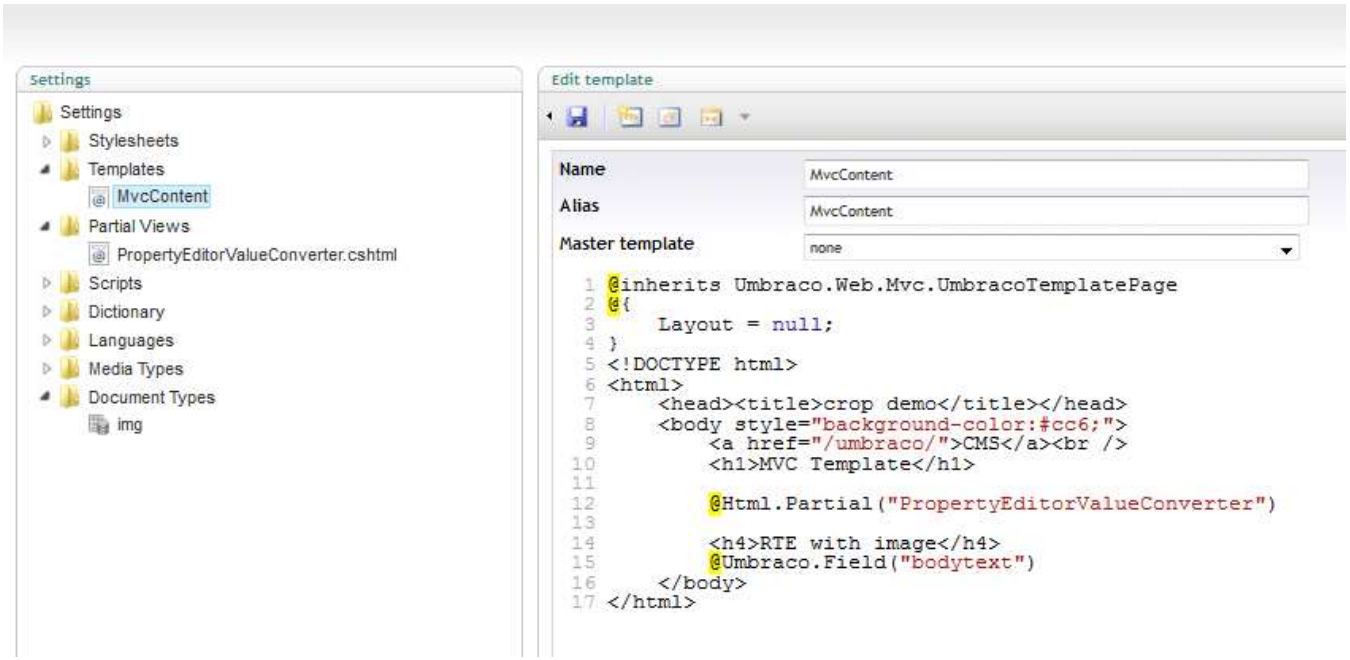
9. Upload another image with the upload, adjust the associated cropper and save & publish the node.



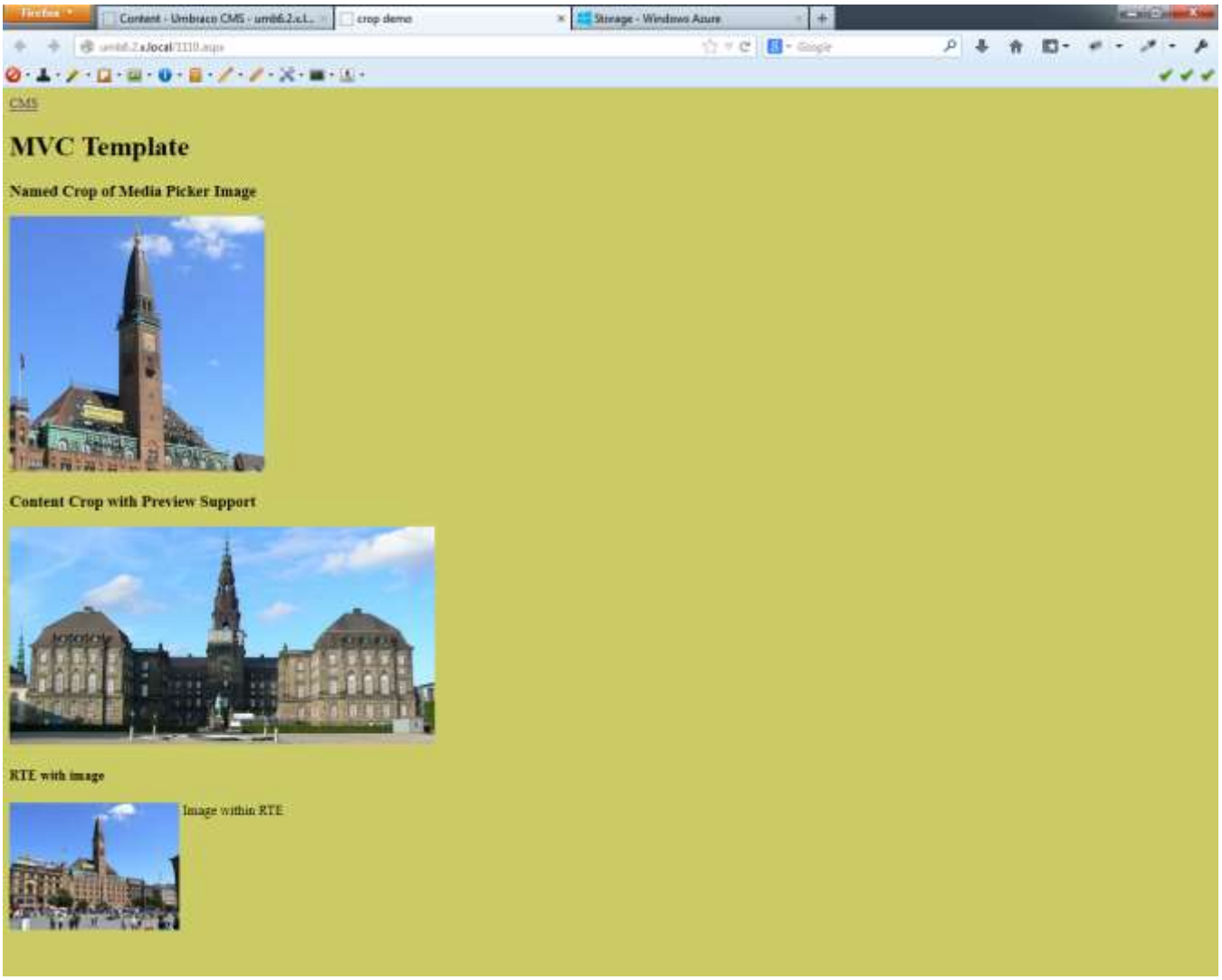
10. Sample scripts for a partial view ...



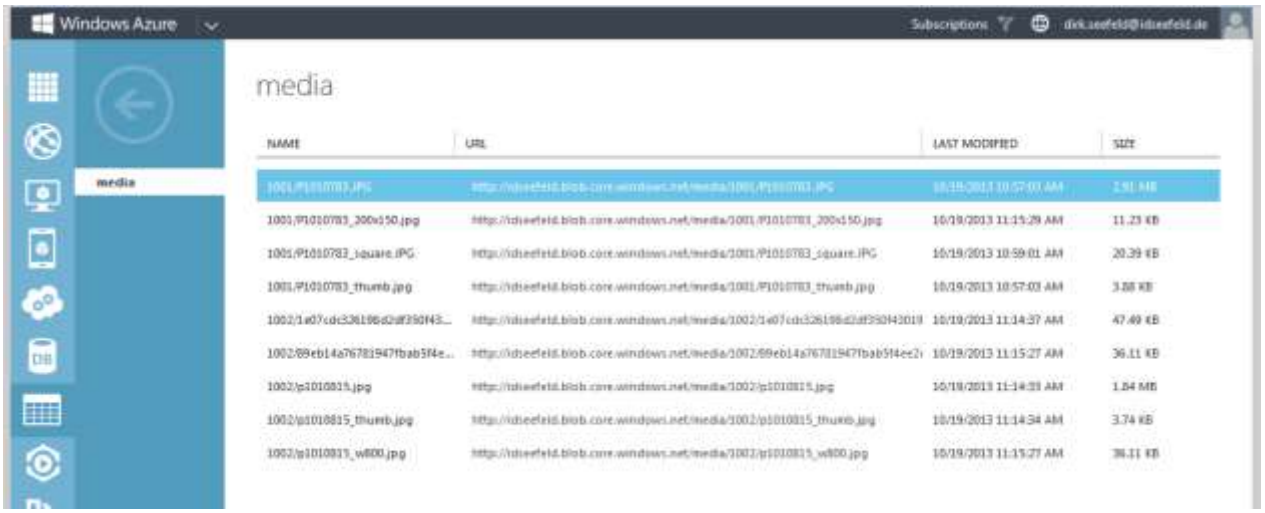
... and one for the MVC template, available in the documentation section of the [package download page](#).



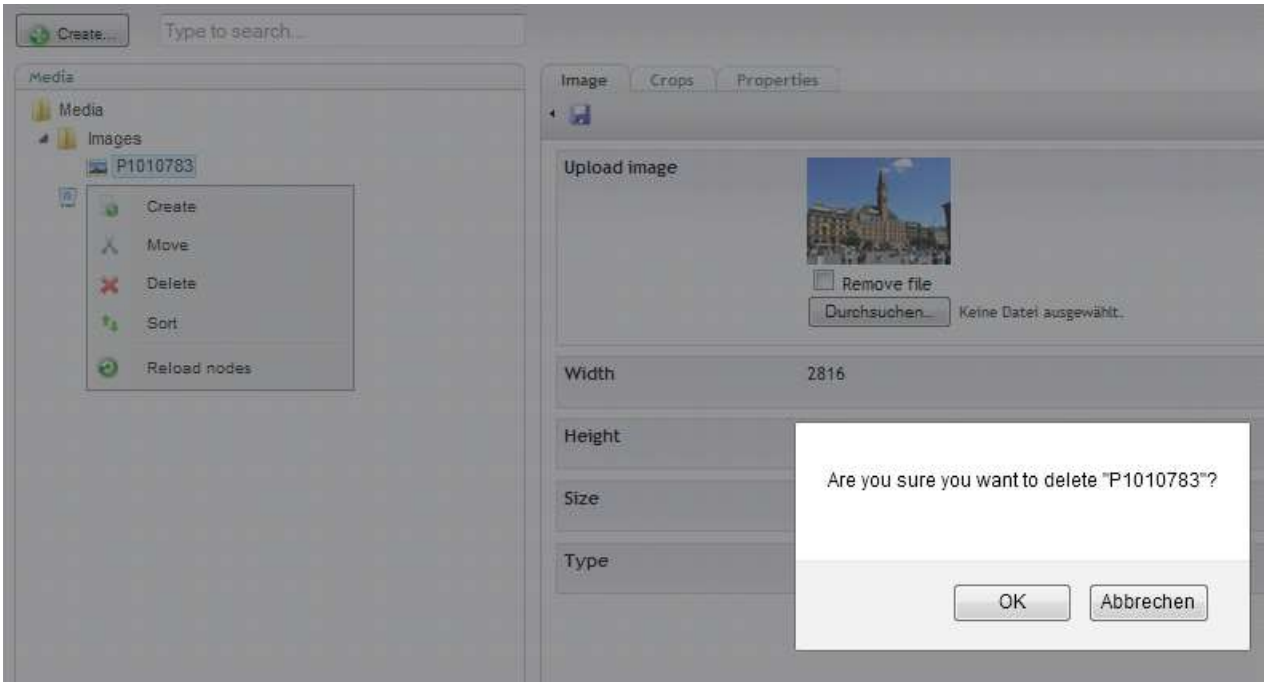
11. Result view of the sample template.

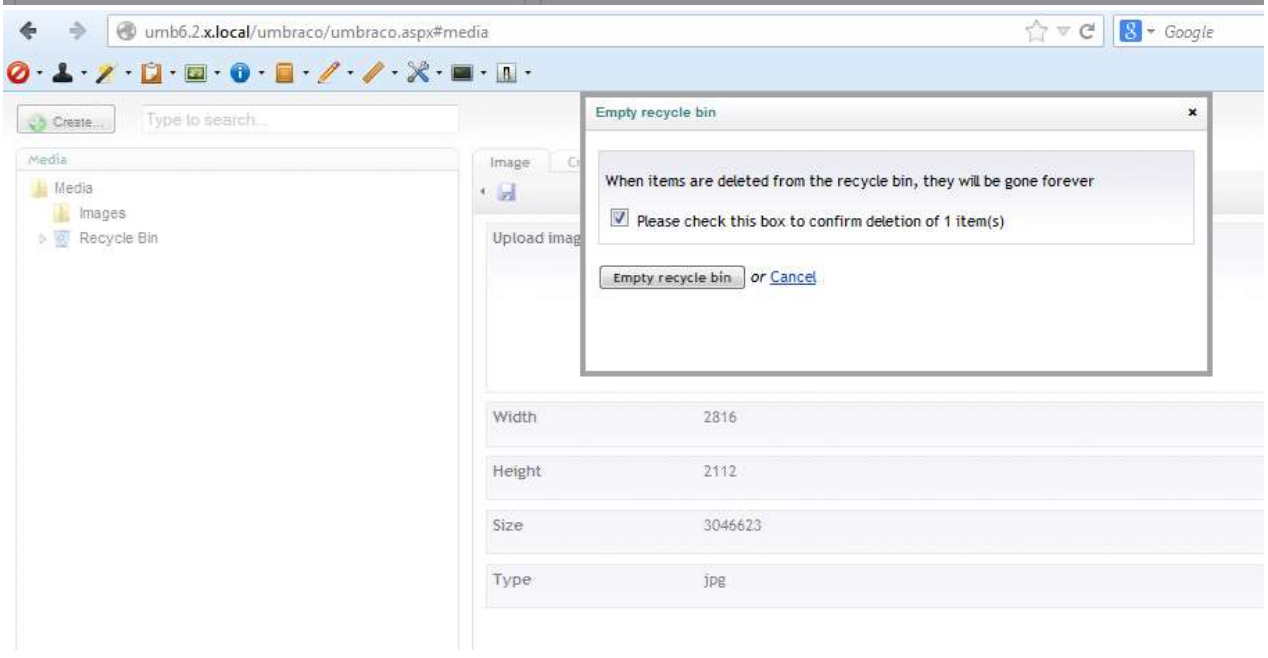


12. List of uploaded files.

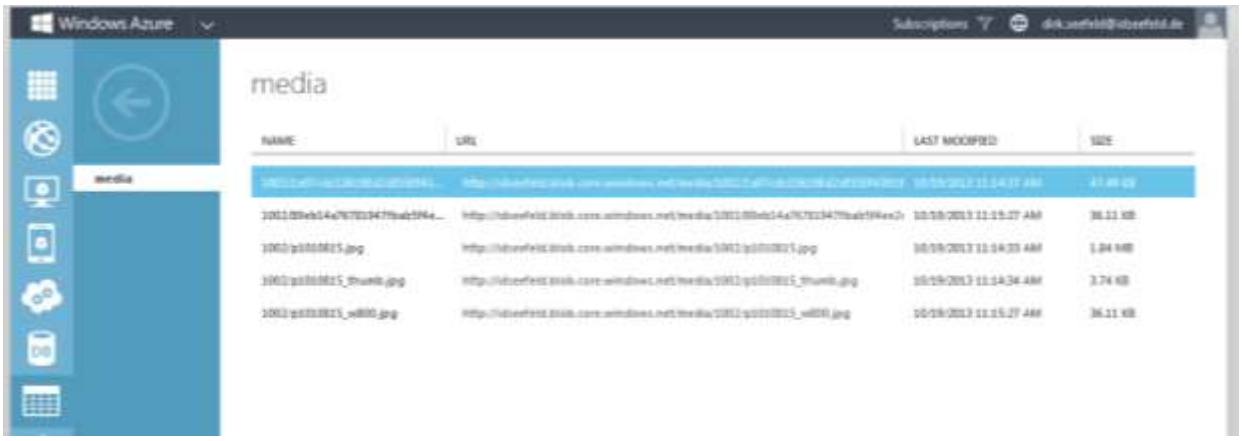


13. Delete a media item.





At the time of this writing (previous to version 6.2.0) there is an issue with deleting a single media item in the recycle bin.
But you can empty the whole recycle bin and all trashed items will be deleted in Azure storage.



An [issue report] with a possible solution is created, but I don't know whether it will make it into the release version 6.2.0 - your vote might help.

14. Bug Fix
    Until version 1.0.5 there was a bug that lead to doubled directory names (actually numbers starting with 1001) for different uploads. In case of deleting an item this actually deletes the whole directory, as this is assumed unique for only one item and siblings like thumbnails for images.
    Version 1.0.8 writes a control file to the repository with the number of the latest existing media directory in case you have data from a previous version. Otherwise, it contains 0. This prevents “old” incorrectly named blobs from deletion. Of course, this is only a work-a-round. However, it is the best solution I could figure out so far.